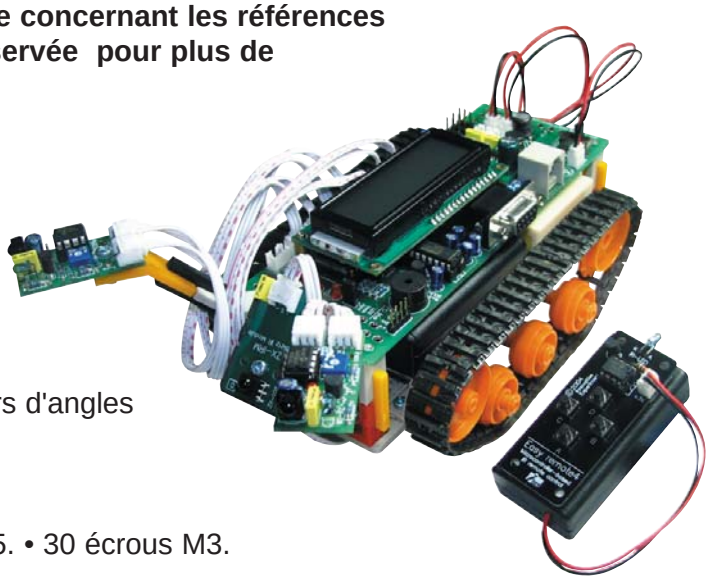
ROBOT EXPÉRIMENTAL contrôlé par PIC
Basé sur le kit ROBO-PICA

Référence : 4005

Cette présente notice se base sur la notice Anglaise concernant les références aux figures ou dessins. La numérotation a été conservée pour plus de compréhension.



1.1 Liste des pièces :

1.1.1 Pièces mécaniques :

- 2 x Motoréducteur 4.5 à 9V/180mA, réduction 48/1
- Couple de 4kg/cm avec leur support.
- 1 x Ensemble de chenilles (G+D) ainsi que les roues d'entrainement, roues libres, axes et supports d'axes.
- 1 x Plaque support et 3 ensembles de 20 connecteurs d'angles différents.

- 1 x Lot de visserie diverse :

• Vis M3 : 4 de 6, 30 de 10, 4 de 15, 4 de 25 et 4 de 35. • 30 écrous M3.
• 3 vis tête plate.
• Entretoises pour vis M3 : 4 de 5, 4 de 10, 4 de 15 et 4 de 25.

1.1.2 Pièce électriques :

- 1 x Carte d'expérimentation RBX-877 pour PIC 16F877.
- 2 x Interrupteur ZX-01.
- 2 x E/R infrarouge ZX-08.
- 2 x Capteur réflectif IR ZX-03.
- 1 x Récepteur de télécommande IR ZX-05.
- 1 x Emetteur de télécommande IR ER-4.
- 1 x Afficheur alphanumérique 2 x 16 caractères.

1.1.3 *D'autres pièces existent en option mais ne sont pas encore disponibles*.

1.2 Outillage nécessaire :

- Pince coupante.
- Cutter.
- Tournevis Philips
- PC équipé de W98SE ou plus avec un port RS232.

1.3 Outils de développement logiciels :

Le kit Robo-PICA utilise un microcontrôleur PIC 16F877 Microchip dans lequel le développeur peut programmer en langage assembleur, en C ou en Basic. Seuls les langages C et Basic doivent utiliser un compilateur.

Cependant, dans ce kit, le seul type de langage utilisé sera le BASIC car celui-ci est facile de compréhension pour les programmeurs débutants.
Le plus populaire des compilateurs BASIC étant le PICBASIC PRO de Melabs, c'est celui-ci qui sera utilisé ici.
La version démo de ce compilateur est disponible sur le site de melabs à l'adresse **http://www.melabs.com/pbpdemo.htm** ou sur le CD joint. Ainsi que le manuel d'utilisation du compilateur.

Produit importé et distribué par :
Selectronic
B.P 10050 - 59891 LILLE Cedex 9
TEL : 0 328 550 328 Fax : 0 328 550 329 SAV : 0 328 550 323 www.selectronic.fr

1.3.1 PicBasic Pro Compiler (version démo)

La version demo du compilateur Melabs est limitée à 31 lignes de programme (les blancs et lignes de remarques ne compte pas) et ne peut pas inclure d'autre fichiers par la commande " INCLUDE ".

La version démo supporte les PIC 16 F 627(A) - 628(A) - 84(A) - 870 - 871- 872 - 873(A) - 874(A) - 876(A) et 877(A). Le compilateur compile le programme basis evec une extension .BAS en un fichier machine en .HEX en deux passe :

- La première compile le fichier .bas en un fichier assembleur .asm.
- La seconde compile le fichier assembleur .asm en un fichier machine .hex.

1.3.1.1 Liste des mots réservé au PicBasic Proc Compiler :

Un certain nombre de mots sont exclusivement réservés au compilateur. L'utilisateur ne peut pas utiliser ces mots pour définir des variables ou des labels.
La liste complète se trouve pages 8 et 9 de la notice en Anglais.

1.3.1.2 Installation de la version démo du compilateur :

Lancer le programme Install après avoir télécharger la version démo du PicBasic Pro Compiler.
Le programme installe la version démo et l'interface Windows MicroCode Studio.

Pour commencer à programmer il suffit de lancer le programme MicroCode Studio dans le menu démarrer.

1.3.1.3 Utilisation du PicBasic Pro Compiler dans un environnement Windows.

Le compilateur PicBasic Pro est un programme fonctionnant sous environnement DOS et pouvant être lancé dans une ligne de commande. Cependant, La société Mecanique a développée une interface Windows appelée MicroCode Studio pour ce compilateur.

1.3.2 MicroCode Studio :

MicroCode Studio est développé par Mecanique UK et est libre d'utilisation. C'est in environnement de développement visuel incluant un débogueur in situ spécifiquement développé pour le compilateur de melabs.

Il est très simple de paramétrer l'assembleur, le compilateur et le programmateur ou de laisser MicroCode Studio le faire à votre place.

1.3.2.1 Installation de MicroCode Studio :

Le programme d'installation de Microcode Studio MCStudio.exe est disponible sur le CD. Il suffit de cliquer sur ce fichier pour lancer l'installation.

Lancer le programme dans le menu " Démarrer " / Programmes / Microcode Studio / Microcode Studio.
Une fois lancé la fenêtre principale s'ouvre (voir figure 1.1).

1.3.2.2 Ajout du compilateur PicBasic Pro :

Lors de son premier lancement, Microcode Studio vérifie automatiquement la présence du compilateur PicBasic Pro. Si vous avez un autre compilateur installé (par exemple la version standard) vous devez lui indiquer manuellement l'emplacement du répertoire.

Pour cela, sélectionner le menu " View " / " PICBasic Options " et dans l'index " Compiler "
(voir figure 1.2) cliquer le bouton " Find Manually " et selectionner le répertoire contenant le
compilateur à utiliser (voir figure 1.3) puis cliquer sur " OK ".

1.3.2.3 Description :

La fenêtre principale de Microcode Studio comporte plusieurs éléments et zones.
Voir Figure 1.4

Basic source code area : cette zone correspond au programme lui-même, La couleur de la police de caractère permet de repérer rapidement les instructions, labels et commentaires afin d'en faciliter la compréhension par le programmeur.

Code Explorer : Cette fenêtre permet un accès aux variables, constantes, définitions, symboles, labels etc.…

Error window : Un certain nombre d'erreur peuvent être généré à la compilation du code source. Le programmateur peut accéder facilement à la ligne concernée en cliquant sur l'erreur.

Status Bar : Cette ligne indique en vert ou en rouge l'état du programme.

Tools Bar : Cette barre comporte les éléments permettant d'éditer, compiler, vérifier etc….

Serial communication Window : Cette fenêtre est utilisée pour afficher les données sérielles envoyées par le microcontrôleur cible par les commandes DEBUG ou SEROUT. L'utilisateur peut définir tous les paramètres.

1.3.2.4 Boutons de la barre de menu :

FILE

• **NEW :** Cette commande ouvre un nouveau document. Une entête est automatiquement générée avec les information d'auteur, de copyright et la date. Pour valider ou non cette fonction et modifier cette entête il suffit de choisir la fonction " Editor Option " lorsqu'un nouveau document est créé. Le contrôleur cible par défaut est le 16F877 et pour le changer if suffit d "éditer et de modifier le fichier default.ini dans le répertoire d'installation. Il est nécessaire de relancer Microcode Studio après toute modification.
• **OPEN :** Commande d'ouverture d'un fichier existant au format de microcode Studio.
• **SAVE :** Sauvegarde du code Basic en cours dans un fichier.
• **SAVE ALL :** Sauvegarde tous les fichiers ouverts.
• **CLOSE :** Ferme le Code Basic en cours.
• **CLOSE ALL :** Ferme tous les documents.
• **REOPEN :** Affiche la liste des derniers fichiers utilisés.
• **PRINT SETUP :** Ouvre la boîte de dialogue d'impression.
• **PRINT PREVIEW :** Affiche l'aperçu d'impression.
• **PRINT :** Imprime la fenêtre active.
• **EXIT :** Fin du programme.

EDIT

• **UNDO :** Retourne en arrière d'un pas.
• **REDO :** Annule la dernière commande Undo.
• **CUT :** Couper
• **COPY :** Copier
• **PASTE :** Coller
• **DELETE :** Effacer
• **SELECT ALL :** Sélectionne tout le texte de la fenêtre active
• **FIND :** Ouvre la boîte de dialogue Rechercher.
• **REPLACE :** Ouvre la boîte de dialogue Rechercher/Remplacer
• **FIND NEXT :** Cherche la prochaine occurrence.

VIEW

- **CODE EXPLORER :** Affiche ou Masque la fenêtre du code.
- **SERIAL COMMUNICATION WINDOW :** Ouvre le logiciel de communication sérielle.
- **TOOLBARS :** Permet de sélectionner les fenêtres actives.
- **Standard :** Outils standard tels que Ouverture/Sauvegarde…..
- **Compile and Program :** Outils de compilation et de programmation.
- **ICD :** Débogage en ligne (mon supporté par la version démo).
- **Edit :** Outils d'édition du code source.
- **PICBASIC OPTION :** Recherche et sélection du répertoire d'installation du programme PICBASIC PRO compiler et du logiciel du programmateur.
- **Compiler :** Sélection le répertoire du compilateur.
- **Assembler :** Sélectionne l'assembleur. Normalement, Microcode Studio sélectionne PM.EXE comme assembleur. Il n'est pas nécessaire de changer cette option.
- **Programmer :** Sélection du répertoire du logiciel du programmateur.

- **EDITOR OPTION :** Ouvre la boîte de dialogue de l'éditeur.
- **General :** Sélection des paramètres généraux.
- **Highlighter :** Réglage des couleurs de polices.
- **Program Header :** Permet de définir l'entête.
- **Online Updating :** Vérifie la mise à jour du logiciel.

1.3.3 Logiciel de programmation Miracle PIC :

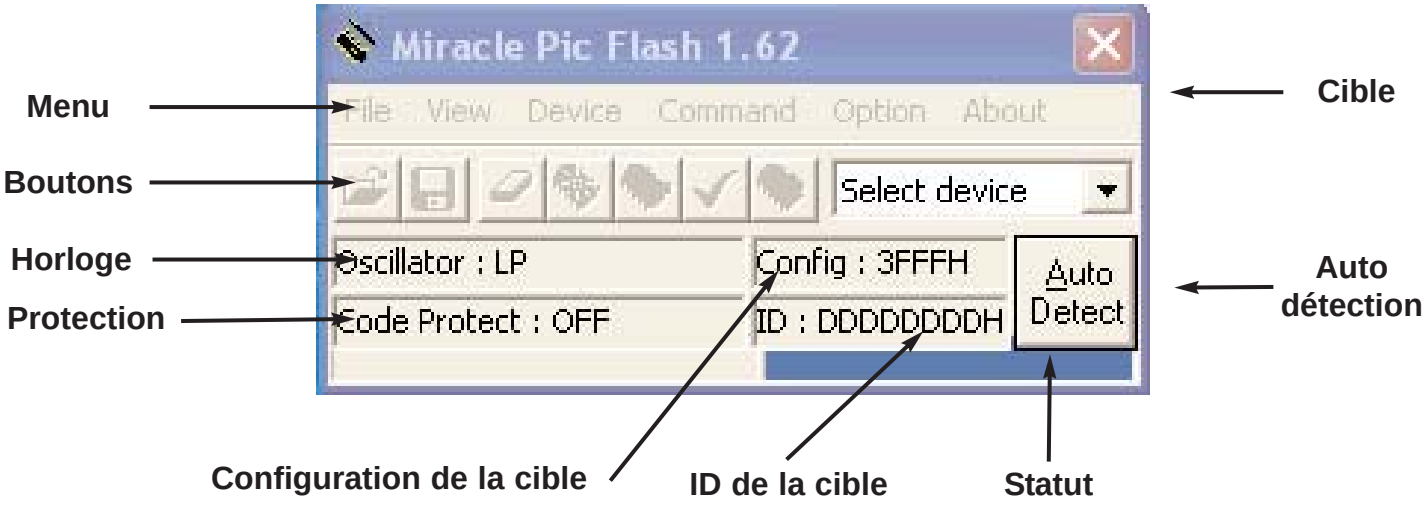
Ce logiciel est libre d'utilisation et est téléchargeable sur le site de Inex ou disponible sur le CD. Ce logiciel à besoin d'un port parallèle libre (LPT) et est reconnu par l'environnement Microcode Studio. L'utilisateur peut compiler le code Basic et programmer le PIC en in click.

1.3.3.1 Installation :

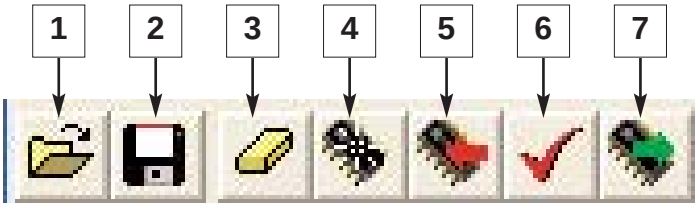
A partir du répertoire Miracle_PIC du CD, lancer le programme mPICSetup161.exe (le numéro de version peut être différent) et cliquer sur OK pour installer le logiciel.

Une fois l'installation terminée, lancer le programme MiraclePIC en le sélectionnant dans le menu "

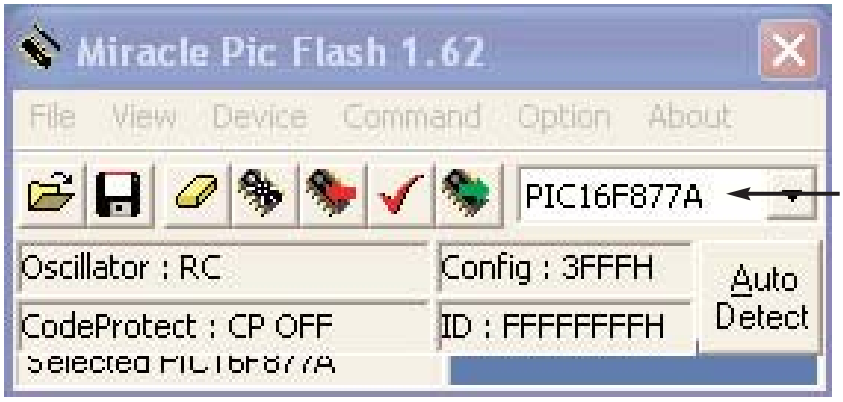
- Démarrer " / " Programmes " / " Miracle PIC "
- La fenêtre principale s'ouvre : Figure 1.5



1.3.3.2 Description des boutons :



1) Open : Ouverture d'un fichier HEX
2) Save : Sauvegarde d'un fichier HEX
3) Erase : Effacement de la cible
4) Blank Check : Test de virginité
5) Program : Programmation de la cible
6) Verify : Vérification de la programmation
7) Read : Lecture de la cible



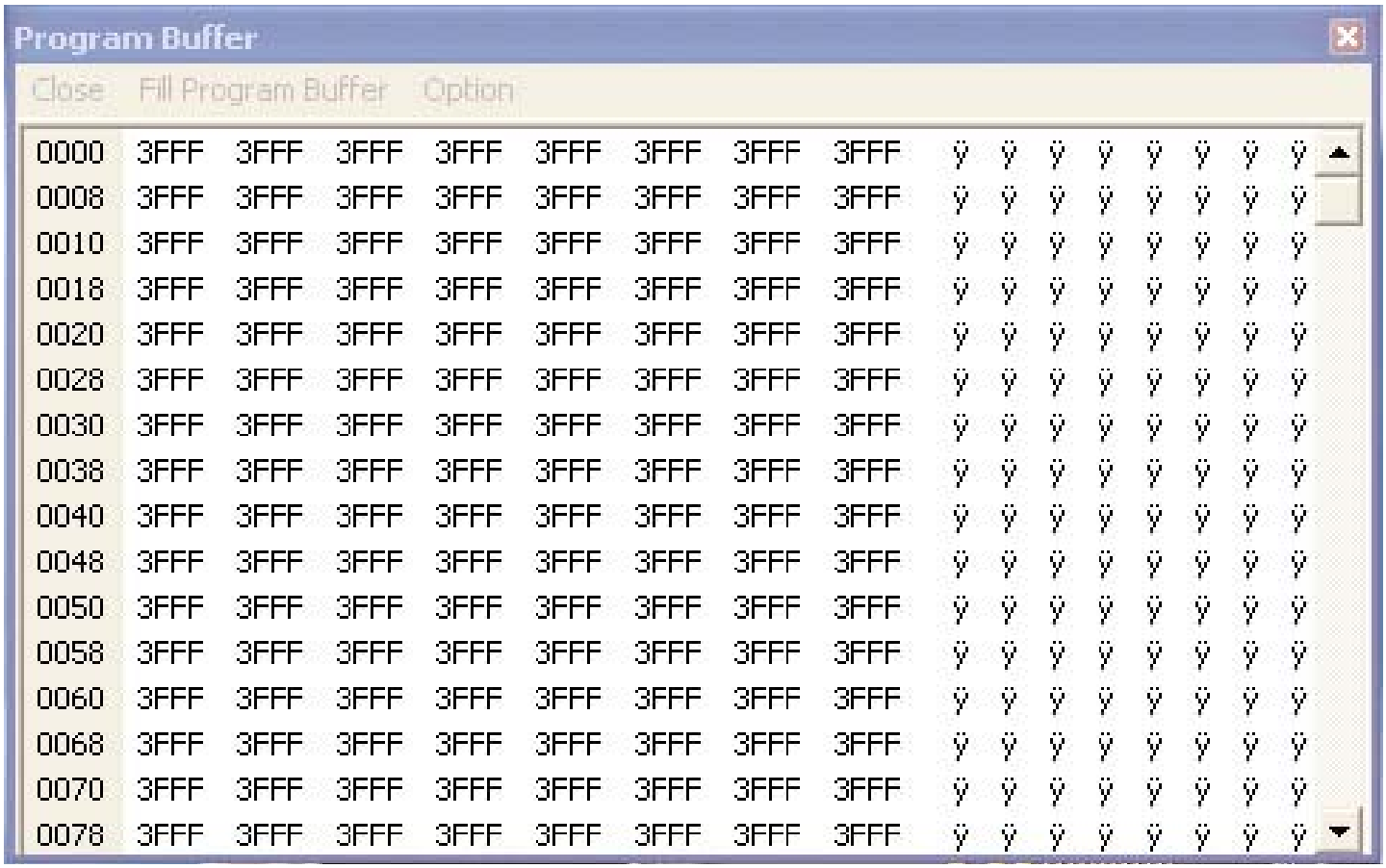
Choix du PIC à utiliser

1.3.3.3 Description des menus :

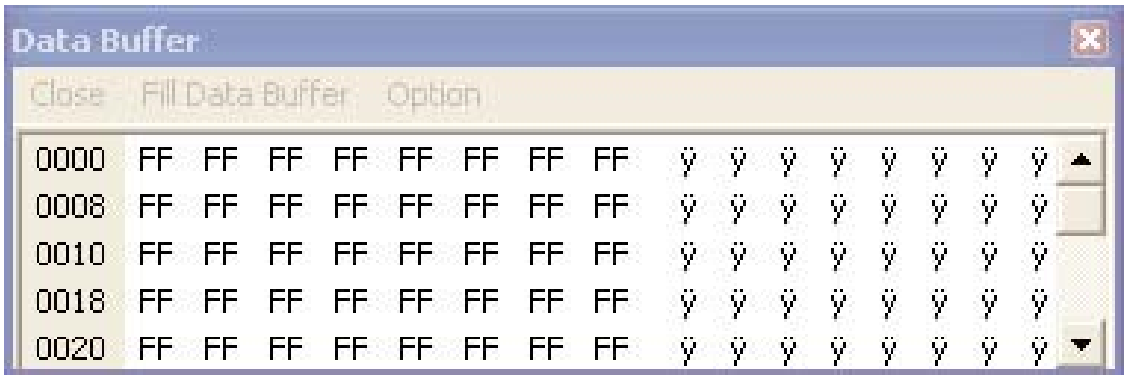
File ***Gestion des fichiers***
Open file : Pour ouvrir un fichier .hex
Save file as : Pour sauvegarder un fichier sous un autre nom
Recent files : Fichiers ouverts récemment
Exit : Sortie du programme

View ***Visualisation de tous les paramètres du PIC.***

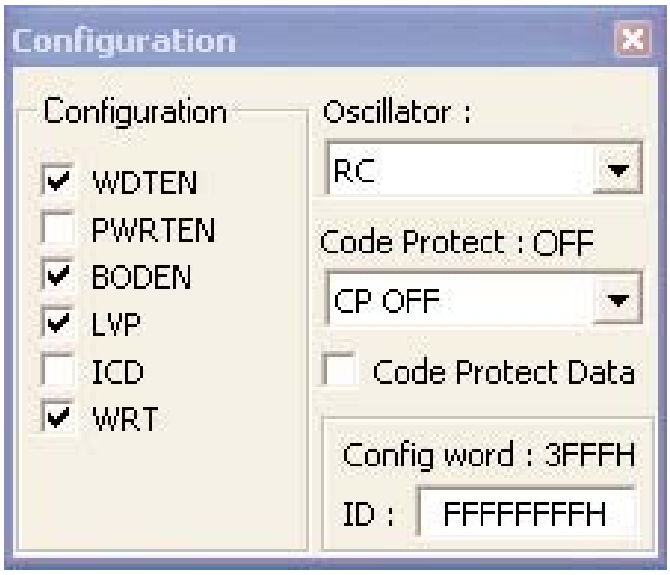
Program Buffer (F11): Affichage du code de la mémoire programme. Chaque position de la mémoire peut être éditée. Figure 1.6



Data Buffer (F12) : Affichage du code de la mémoire donnée (EEPROM).
Chaque position de la mémoire peut être éditée. Figure 1.7


Data Buffer
Close
Fill Data Buffer
Option
0000 FF FF FF FF FF FF FF FF
0008 FF FF FF FF FF FF FF FF
0010 FF FF FF FF FF FF FF FF
0018 FF FF FF FF FF FF FF FF
0020 FF FF FF FF FF FF FF FF

Configuration (F9) : Affichage des paramètres de configuration du PIC.
Chaque paramètre peut être validé ou non.



Count Program : Affiche le nombre de contrôleur programmés (cette fonction n'est pas utilisée avec la carte JX-877)

Configuration setting ***Configuration du PIC avant de le programmer***

Oscillator : Sélection du mode pour l'oscillateur du PIC
- LP - Quartz basse fréquence (non utilisé pour JX-877)
- XT - Quartz standard inférieur à 4MHz (non utilisé pour JX-877)
- HS - Quartz haute fréquence supérieur à 4MHz
- RC - Circuit RC externe (non utilisé pour JX-877)

Code protection : Utilisé pour sélectionner le mode de protection du programme et des données.

Configuration : Utilisé pour sélectionner les fonctions spéciales du PIC
- WPT - Watchdog Timer
- PWRT - Power-up Timer
- BOBEN - Brown-out Reset
- LVP - Low Voltage Program (programmation des PIC en 5V au lieu de 12V non applicable avec JX-877)
- ICD - In-circuit Debugger (non applicable avec JX-877)
- WRT - Flash program Write

Device : ***Sélection du PIC cible***
Command : ***Lecture, écriture et test de la cible***

Program Chip (F4) - Programmation du PIC avec les valeurs des tampons programme et données.
Verify Chip (F5) - Contrôle la programmation du PIC avec les tampons
Read Chip (F6) - Lecture des zones programme et données du PIC
Erase Chip (F7) - Effacement du contenu du PIC
Blank Chip (F8) - Test de virginité

Option

Program/verify Program - Vérification de la zone programme en fonction du tampon programme
Program/Verify Data - Vérification de la zone données en fonction du tampon données
Program/verify ID - Vérification de l'ID du PIC (non applicable avec JX-877)
Program/verify configuration - Vérification de la configuration du PIC
Verify While program - Vérification octet à octet lors de la programmation
Auto verify after program - vérification automatique à la fin de la programmation
Auto erase before program - Effacement du PIC avant programmation
Program all address - Programmation de toutes les adresses (même non alouées)
Configuration erase message - Sélection de la fenêtre d'avertissement
Parallel port - Sélection du port parallèle LPT1 ou LPT2

1.3.3.4 Interfaçage avec Microcode Studio :

Ouvrir Microcode Studio et sélectionner " VIEW " dans la barre de menu puis sélectionner l'onglet " Programmer "

Cliquer sur " Add New Programmer ". La fenêtre d'ajout de programmateur s'ouvre.

Cocher la case " Create a custom programmer " puis " Next ". Une fenêtre s'ouvre pour définir le nom du programmateur : Entrer "Miracle PIC" puis " Next ".

Sélectionner le nom du programme associé soit mPICFlash.exe pour Miracle PIC puis " Next ".

Rechercher le répertoire d'installation du programme manuellement ou laisser Microcode Studio le chercher automatiquement.

Sélectionner les paramètres d'échange entre Microcode Studio et Miracle PIC :

-e -I $hex-filename$ -p -v -x

-e	: Efface les données de la zone programme avant la programmation.
-I $hex-filename$	: charge le fichier cible .hex à partie de Microcode Studio.
-p	: Programme le PIC.
-v	: Vérifie le PIC après programmation.
-x	: Sort du programme à la fin de la programmation.

Cliquer sur " Finish " pour terminer.

2) Carte d'expérimentation RXB-877

Le kit Robo-PICA utilise la carte RXB-877 comme carte de contrôle. Celle-ci est équipé d'un PIC 16F877 comme Microcontrôleur. La Figure 2.1 représente le diagramme interne avec les différents blocs composants cette carte.

La figure 2.2 donne la disposition des connecteur correspondants et la figure 2.3 le schéma de la carte.

2.1 Caractéristiques techniques :

• 16F877(A) cadencé à 20MHz avec 8kmots de mémoire flash.
• Téléchargement via le port Parallèle (sélectionnable par un interrupteur).
• Port de communication sériel. • Interface pour afficheur LCD 2x16. • HP piezo. • 3 LEDs de contrôle.
• Interface de puissance pour 2 moteur DC 4.5 à 6V. • Interface pour 3 servos RC 4.5 à 6V.
• 8 lignes de port programmables comme E/S digitales ou analogiques.
• Alimentation par 4 piles ou accus taille AA (Accus 1.7Ah recommandés).

2.2 Description de la carte RBX-877 :

2.2.1 Microcontrôleur :

Le cœur de cette carte est un PIC 16F877(A) de chez Microchip cadencé à 20MHz par un résonateur céramique.

2.2.2 Alimentation :

La carte RBX-877 comporte un régulateur à découpage permettant de fournir du 5V au microcontrôleur même lorsque la tension d'accus descend à 1.5Vmin. (Par exemple lorsque les moteurs tournent).

L'interrupteur S1 permet de mettre en service ou non l'alimentation et la led BATT permet de connaître l'état des accus.

2.2.3 Programmation in situ :

Afin de pouvoir satisfaire au mode de programmation in situ, la carte comporte une deuxième alimentation à découpage fournissant du 12.5V. L'interrupteur MODE permet de choisir entre Programmation (Led PGM) ou Utilisation (Led RUN).

2.2.4 Afficheur, Leds et Buzzer:

Un connecteur pour afficheur 2 x 16 caractères est disponible sur la carte. Il est câblé sur les ports RD4-RD7 pour les données D4-D7, RC5 pour la ligne E et RC0 pour la ligne RS. L'ajustable à coté du connecteur Sub-D permet de régler le contraste de l'afficheur et le cavalier à coté des Leds rouge permet d'alimenter ou non le rétro-éclairage.
D'autre part, 3 Leds rouge sont câblées sur les lignes de port RB3-RB5 avec une résistance de limitation de courant.

Un HP piezo de bande passante 1-3kHz est câblé sur la ligne de port RA4.

2.2.5 Port d'E/S :

La carte RBX-877 dispose de 8 ports d'E/S programmables incluant RA0-RA3, RA5 et RE0-RE2. Ces ports sont disponibles sur des connecteurs 3 points incluant le +5V et la masse.

2.2.6 Port série RS232 :

Un port de communication RS232 bidirectionnel est disponible sur un connecteur SubD-9F. Il est câblé sur les lignes de port RC6 et RC7 correspondant à l'UART du PIC via un MAX232.

2.2.7 Sorties Moteurs :

2 sorties pour moteur courant continu 4.5/6V, 100/200mA sont disponibles. Le moteur A est contrôlé par RD0 et RD1 pour la vitesse et RC2 pour la validation. Le moteur B par RD2 et RD3 pour la vitesse et RC1 pour la validation. 2 Leds bicolores permettent de connaître l'état de la sortie moteur correspondante. La tension de commande des moteurs est directement prélevée sur l'entrée d'alimentation.

2.2.8 Sorties Servos RC :

3 sorties pour servos de radio-commande (4.8/6V, 100/200mA) sont disponibles sur les lignes de port RB0-RB2. La tension de commande des servos est directement prélevée sur l'entrée d'alimentation.

2.2.9 Bus I²C :

Un bus I²C est disponible sur un connecteur 4 pts incluant SDA, SCL, la masse et le +5V. Il est connecté sur les lignes de port correspondantes du PIC soit RC3/SCL et RC4/SDA. Ces deux lignes sont munies de résistance de rappel au +5V.

Tous les exemples de code source sont prévus pour le PIC16F877 et donne comme ligne de code de paramétrage :

```
@ DEVICE PIC16F877 , HS_OSC  ' Use PIC16F877 and HS Oscillator
```

En cas d'utilisation d'un PIC 16F877A ou d'un autre PIC il faudra impérativement éditer et modifier cette ligne en fonction du PIC choisis